

Requested Patent: JP9091157A

Title: TASK-DEVICE DRIVER COMMUNICATION CONTROL METHOD ;

Abstracted Patent: JP9091157 ;

Publication Date: 1997-04-04 ;

Inventor(s): MOCHIDA MUNEAKI; MURANAKA TOSHIBUMI ;

Applicant(s): OKI ELECTRIC IND CO LTD ;

Application Number: JP19950246094 19950925 ;

Priority Number(s): ;

IPC Classification: G06F9/46 ;

Equivalents: ;

**ABSTRACT:**

**PROBLEM TO BE SOLVED:** To surely perform communication between the plural tasks of the equal priority and a device driver at high speed. **SOLUTION:** The 1st task investigates the state of a 2nd task occupation flag 2 by turning on a 1st task occupation flag 1 and when the 2nd task occupation flag 2 is turned on, control authority is applied to the other task by turning off the 1st task occupation flag. When the 2nd task occupation flag is turned off, the 1st task writes data in a data table and updates a task write pointer. When all the data are completely written in the data table, the 1st task occupation flag 1 is turned off, and the control authority of its own central processing unit is abandoned. A DSP driver reads data from a transmission data table until the contents of the write pointer get equal with the contents of a read pointer.

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平9-91157

(43)公開日 平成9年(1997)4月4日

(51)Int.Cl.<sup>9</sup>

G 0 6 F 9/46

識別記号

3 4 0

庁内整理番号

F I

G 0 6 F 9/46

技術表示箇所

3 4 0 B

3 4 0 C

審査請求 未請求 請求項の数1 O L (全 8 頁)

(21)出願番号 特願平7-246094

(22)出願日 平成7年(1995)9月25日

(71)出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72)発明者 持田 宗明

東京都港区虎ノ門1丁目7番12号 沖電気  
工業株式会社内

(72)発明者 村中 俊文

東京都港区虎ノ門1丁目7番12号 沖電気  
工業株式会社内

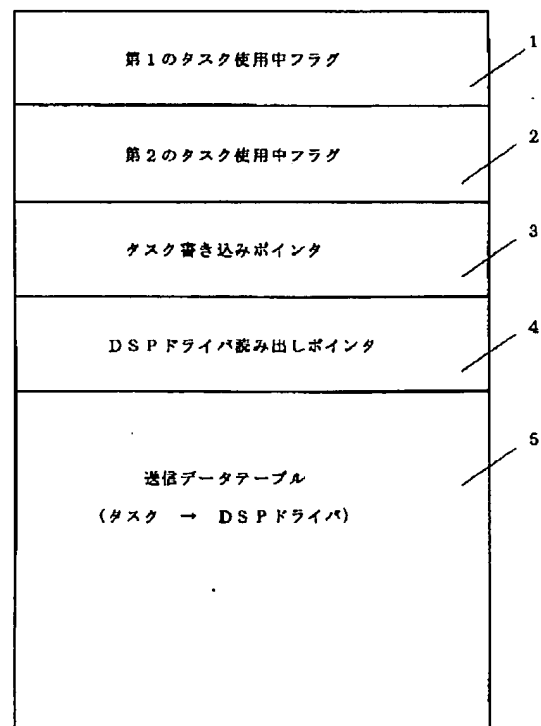
(74)代理人 弁理士 工藤 宣幸

(54)【発明の名称】 タスクデバイスドライバ間通信制御方法

(57)【要約】

【課題】 同一プライオリティの複数のタスクとデバイスドライバの間の通信をを高速且つ確実に行う。

【解決手段】 第1のタスクは、第1のタスク使用中フラグをONにして、第2のタスク使用中フラグの状態を調べ、第2のタスク使用中フラグがONのときは、第1のタスク送信中フラグをOFFにして、他のタスクに制御権を与える。第2のタスク送信中フラグがOFFのときは、第1のタスクは、データテーブルにデータを書き込み、タスク書き込みポイントを更新する。全てのデータをデータテーブルに書き込み終えると、第1のタスク使用中フラグをOFFにして自ら中央処理装置の制御権を放棄する。DSPドライバは、書き込みポイントと読み出しポイントの内容が等しくなるまで、送信データテーブルからデータを読み出す。



## 【特許請求の範囲】

【請求項1】 周辺デバイス及びこの周辺デバイスに対応するデバイスドライバを備えた並列処理システムにおけるデバイスドライバと同一プライオリティの複数のタスクとの間の通信を制御するタスクーデバイスドライバ間通信制御方法において、同一プライオリティの複数のタスクから上記デバイスドライバへの、及び又は上記デバイスドライバから同一プライオリティの複数のタスクへの送信データを蓄積するテーブルを設けて、同一プライオリティの複数のタスクの前記テーブルに対するアクセス要求をフラグにより調整し、アクセス権を取得したタスクが前記テーブルに対してデータを書き込み、及び又は前記テーブルから読み出し、デバイスドライバが前記テーブルからデータを読み出す、及び又は前記テーブルに対してデータを書き込むことを特徴とするタスクーデバイスドライバ間通信制御方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、デバイスドライバと同一プライオリティの複数のタスクとの間の通信を制御するタスクーデバイスドライバ間通信制御方法に関し、例えば、携帯電話における呼処理モジュールに適用し得るものである。

## 【0002】

【従来の技術】近年の携帯電話等の無線端末では、端末の処理能力向上及び小型化のために、音声等のデジタル処理にデジタルシグナルプロセッサ（以下の記述においてDSPと呼ぶ）が使用されており、DSPは呼処理段階に応じて所定の呼制御タスクから適宜アクセスされる。

【0003】従来、携帯電話等の並列処理システムでは、呼制御タスク（アプリケーションタスク）がDSPドライバ（デバイスドライバ）に対してデータを送信する場合に以下のような処理が行われていた。ここで、タスクとは、中央処理装置のスケジュール対象となる並列処理単位のことをいい、一般的にはタスク、スレッド、プロセスと呼ばれているものである。また、デバイスドライバとは、アプリケーションタスクの発行した入出力要求を受けて周辺デバイスを制御するためのプログラムであり、データとサブルーチンからなり、アプリケーションタスクと周辺デバイスとのインターフェースを構成し、周辺デバイスでのイベント発生による割り込み要求を受けて割り込みハンドラを呼び出して処理を行うタスク又はハンドラである。

【0004】第1の呼制御タスクが実行中で、第2の呼制御タスクが中央処理装置の制御権の割り当てを待っているときに、第1の呼制御タスクがDSPドライバに対してデータを送信しようとする。ただし、両呼制御タス

クのプライオリティは同じである。

【0005】このとき、第1の呼制御タスクは、自己のプライオリティを上げることより、第2の呼制御タスクの動作を中断させる。プライオリティが上がった第1の呼制御タスクは、メールを用いてDSPドライバにデータを送信する。ここで、メールとは、任意のタスクからの送信データを待ち行列に登録し、この待ち行列から任意のタスクがデータを受信する、タスク間のデータ送受信方法である。

【0006】DSPドライバへのデータ送信が終了すると、第1の呼制御タスクが自ら中央処理装置の制御権を放棄することにより、第2の呼制御タスクは、中央処理装置の制御権を獲得し、処理を再開する。

## 【0007】

【発明が解決しようとする課題】前述したように、従来は携帯電話の高速な処理の要求される呼処理ファームにおいてもメールを使用したタスク間のデータのやりとりが行われていて、メールによるデータ送信はプライオリティの違うタスクからのデータ送信も考慮していたために処理が複雑になり、アプリケーションタスクとデバイスドライバ間のデータ送信速度は多大な制約を受けていた。さらに、タスクのプライオリティを上げることにより、他のタスクの処理の停止あるいは処理速度の低下を招くことになり、システム全体の処理速度が低下した。

【0008】また、アプリケーションタスクのプライオリティを上げずに、アプリケーションタスクからデバイスドライバに対してデータ送信した場合は、他のアプリケーションタスクにより送信データが書きかえられる可能性があった。

【0009】本発明は、以上の点を考慮してなされたものであり、携帯電話の呼処理モジュールのように高速な処理の要求されるモジュールにおいて、同一プライオリティの複数のアプリケーションタスクとデバイスドライバ間のデータのやりとりを（特に、同一プライオリティの複数のアプリケーションタスクから、高速に処理を完了する必要があるデバイスドライバへのデータの送信を）、任意のタイミングで高速に且つ確実に行うタスクーデバイスドライバ間通信制御方法を提供しようとしたものである。

## 【0010】

【課題を解決するための手段】本発明は、周辺デバイス及びこの周辺デバイスに対応するデバイスドライバを備えた並列処理システムにおけるデバイスドライバと同一プライオリティの複数のタスクとの間の通信を制御するタスクーデバイスドライバ間通信制御方法において、以下のようにしたことを特徴とする。

【0011】すなわち、同一プライオリティの複数のタスクから上記デバイスドライバへの、及び又は上記デバイスドライバから同一プライオリティの複数のタスクへの送信データを蓄積するテーブルを設けて、同一プライ

オリティの複数のタスクの前記テーブルに対するアクセス要求をフラグにより調整し、アクセス権を取得したタスクが前記テーブルに対してデータを書き込んでからデバイスドライバが前記テーブルからデータを読み出す、及び又はデバイスドライバが前記テーブルに対してデータを書き込んでからアクセス権を取得したタスクが前記テーブルから読み出す。

【0012】送信データを蓄積するテーブルを備えることにより、タスクとデバイスドライバの間のデータのやりとりを高速に行い、同一プライオリティの複数のタスクの前記テーブルに対するアクセス要求をフラグにより調整することで、同一プライオリティの複数のタスクからデバイスドライバへのデータ送信要求に対する実行を制御すると共に、デバイスドライバから同一プライオリティの複数のタスクへのデータ送信においても同一プライオリティの複数のタスクの実行を制御する。

【0013】

【発明の実施の形態】以下、本発明による、デバイスドライバと同一プライオリティの複数のタスクとの間でデータの送受信を行うタスク-デバイスドライバ間通信方法の実施の形態を、図面を参照しながら詳述する。

【0014】(第1の実施の形態)本実施の形態は、本発明を無線方式移動携帯電話における呼制御タスクからDSPドライバへのデータ送信に適用したものである。

【0015】図2に本実施の形態のタスク-デバイスドライバ間通信方法を無線方式移動携帯電話に適用したときの概念を示す。タスク監視プログラム25は、実行中のタスクが終了したとき、又は実行中のタスクが自ら図示しない中央処理装置の制御権を放棄したとき、又は割り込み要求があったときに、システム内のタスクの状態を調べて、最もプライオリティの高いタスクに中央処理装置の制御権を与え、当該タスクを実行状態にする。呼制御タスク21及び22は同じプライオリティをもち、DSPドライバ24にデータを送信する。DSPドライバ24は、呼制御タスク21及び22からデータを受信して、DSPを制御する。インターフェーステーブル23は、図1に示すように、第1及び第2のタスク送信中フラグ1及び2、タスク書き込みポイント3、DSPドライバ読み出しポイント4、並びに送信データテーブル5からなっている。第1及び第2のタスク使用中フラグ1及び2はそれぞれ、ON又はOFFの何れかの状態にある。

【0016】このような状態で、第1の呼制御タスク1がDSPドライバ24にデータを送信しようとしたときの処理の流れを図3を参照しながら説明する。

【0017】第1の呼制御タスク21は、第1のタスク使用中フラグ1をONにして、第2のタスク使用中フラグ2の状態を調べ、第2のタスク使用中フラグ2がONのときは、第1のタスク送信中フラグ1をOFFにして、自ら中央処理装置の制御権を放棄し、タスク監視プ

ログラム25が他のタスクに制御権を与える。(ステップ301~304)。

【0018】第2のタスク送信中フラグ2がOFFのときは、第1の呼制御タスク21は、送信データテーブル5に一定量のデータを書き込み、タスク書き込みポイント3を更新する。全てのデータを送信データテーブル5に書き込むまでこの動作を繰り返す(ステップ305~307)。

【0019】第1の呼制御タスク21は、全てのデータを送信データテーブル5に書き込み終わると、第1のタスク使用中フラグ1をOFFにして自ら中央処理装置の制御権を放棄する(ステップ308)。

【0020】以上のように、第1及び第2のタスク送信中フラグ1及び2の状態を管理することで、呼制御タスク21及び22の処理実行を制御する。

【0021】続いて、DSPドライバ24が送信データテーブル5に蓄積されたデータを読み出すときの処理の流れを図4を用いて説明する。

【0022】DSPドライバ24は、タスク書き込みポイント3とDSPドライバ読み出しポイント4の内容が等しくなるまで、送信データテーブル5からの一定量のデータの読み出しとDSPドライバ読み出しポイント4の更新を繰り返し、双方のポイント3及び4の内容が等しくなったら処理を終える(ステップ401~403)。

【0023】このように送信データテーブル5を使用することで、メールを使用することなく、呼制御タスク21又は22からDSPドライバ25へのデータの送信を行う。

【0024】また、第1の呼制御タスク21が送信データテーブル5をアクセスしているときにタスクディスパッチが発生したときの処理の流れを図5を用いて説明する。

【0025】第1の呼制御タスク21が第1のタスク使用中フラグ1をONにしたあとで、タスクディスパッチが発生して、第2の呼制御タスク22が動作を開始し、第2の呼制御タスク22は第2のタスク使用中フラグ2をONにして、第1のタスク使用中フラグ1の状態を調べる(ステップ501~503)。

【0026】第1のタスク使用中フラグ1がONであるので、第2の呼制御タスク22は第2のタスク使用中フラグ2をOFFにして、中央処理装置の制御権を放棄する。これによりタスクディスパッチが発生して、第1の呼制御タスク21は再び動作を再開する(ステップ504~506)。

【0027】さらに、第1の呼制御タスク21が送信データテーブル5をアクセスしているときにタスクディスパッチが発生して第2の呼制御タスク22が動作を開始してフラグチェックを終える前に再びタスクディスパッチが発生したときの処理の流れを図6を用いて説明す

る。

【0028】第1の呼制御タスク21が第1のタスク使用中フラグ1をONにしたあとで、タスクディスパッチが発生して、第2のタスクが動作を開始し、第2の呼制御タスク22は第2のタスク使用中フラグ2をONにする(ステップ601, 602)この直後に、再びタスクディスパッチが発生して、第1のタスクが動作を開始する。第1の呼制御タスク21が、送信処理を終えて、中央処理装置の制御権を放棄すると、タスクディスパッチして第2の呼制御タスク22が再び動作を開始する。第2の呼制御タスク22は第1のタスク使用中フラグ1を調べて、後続処理を行う(ステップ603~607)。

【0029】このように、タスクディスパッチが発生するタイミングに関係なく、タスク使用中フラグ1及び2を用いて、呼制御タスク21及び22の処理実行を制御することができる。

【0030】また、DSPドライバ24が、第1の呼制御タスク21が送信データテーブル5に書き込んだデータを読み終える前に第2の呼制御タスク22が送信データテーブル5に書き込みを開始しても、第2の呼制御タスク22は、タスク書き込みポイント3を参照して第1の呼制御タスク21からのデータの後ろに自己の送信データを書き込むことで、データの上書きも発生しない。

【0031】以上の第1の実施の形態によれば、呼制御タスクからDSPドライバへのデータ送信時に、送信データテーブル介して送信を行い、タスク使用中フラグを用いて呼制御タスクの処理実行を調整するようにしたので、高速且つ確実なデータ送信を行うことができる。

【0032】(第2の実施の形態)本実施の形態は、本発明を無線方式移動携帯電話における呼制御タスクのDSPドライバからのデータ受信に適用したものである。インターフェーステーブル4以外は第1の実施の形態と同じであるので、その説明は省略する。

【0033】本実施の形態におけるインターフェーステーブル4のは、図7に示すように受信タスク識別用フラグ71、DSPドライバ書き込みポイント72、タスク読み出しポイント73、及び受信データテーブル74からなる。さらにインターフェーステーブル4は、第1及び第2の呼制御タスクに対応させて、2セット用意する。一方のインターフェーステーブル4内の受信タスク識別用フラグ71は、第1の呼制御タスクに対応していて、常に'1'の値を持ち、同様に他方のインターフェーステーブル4内の受信タスク識別用フラグ71は、第2の呼制御タスクに対応していて、常に'2'の値を持つ。

【0034】このような状態で、DSPドライバ24が第1の呼制御タスク21にデータを送信しようとしたときの処理の流れを図8を参照しながら説明する。

【0035】DSPドライバ24は、受信タスク識別フラグが'1'の値を持つインターフェーステーブルに対

して以下の処理を行う。

【0036】DSPドライバ24は、受信データテーブル74への一定量の送信データの書き込み処理とDSPドライバ書き込みポイント72の更新の処理を、全送信データを書き終えるまで繰り返す(ステップ801~803)。

【0037】続いて、上記の処理で受信データテーブル74に書き込んだデータを第1の呼制御タスク21が読み出す処理を図9を参照しながら説明する。

【0038】第1の呼制御タスク21は、受信タスク識別フラグが'1'の値を持つインターフェーステーブルに対して以下の処理を行う。

【0039】第1の呼制御タスク21は、DSPドライバ書き込みポイント72とタスク読み出しポイント73の内容が等しくなるまで、受信データテーブル74からの一定量のデータの読み出しとタスク読み出しポイント73の更新を繰り返す(ステップ901~903)。

【0040】このように、受信タスク識別フラグ71の値により、タスクの処理実行を制御できる。

【0041】以上の第2の実施の形態によれば、DSPドライバからのデータ受信時に、受信データテーブル介して受信を行い、タスク受信用フラグを用いて呼制御タスクの受信データテーブルに対するアクセスを制御するようにしたので、高速且つ確実なデータ受信を行うことができる。

【0042】(他の実施の形態)第1及び第2の実施の形態では同一プライオリティのタスク(呼制御タスク)が2つのときのタスク-デバイスドライバ間通信方法を説明したが、本発明は、同一プライオリティのタスクが3つ以上あっても同様に処理できる。

【0043】また、本発明はDSPドライバ以外にも、シリアル(RS232C, RS422等)、パラレル、GP-IB、PCMCIA等のインターフェースデバイスドライバにも適用できる。

【0044】さらに、本発明は無線方式移動携帯電話以外にも、デバイスドライバと同一プライオリティの複数のタスクとの間でデータをやりとりする並列処理システムであれば、例えば、電子交換機、ファクシミリ等にも適用できる。

【0045】

【発明の効果】以上説明したように、本発明によれば、デバイスドライバと同一プライオリティの複数のタスク間でデータのやりとりをするときに、送信データを蓄積するテーブルを設けて、同一プライオリティの複数のタスクの前記テーブルに対するアクセス要求をフラグにより調整するようにしたので、同一プライオリティの複数のタスクとデバイスドライバの間のデータのやりとりを高速且つ確実に且つ簡易に行うことができ、並列処理システムの処理能力の向上を低コストで実現できる。

【図面の簡単な説明】

【図1】第1の実施の形態におけるインターフェーステーブルを示す図である。

【図2】第1の実施の形態の概念を示す図である。

【図3】第1の実施の形態でのデータの書き込み処理を示すフローチャートである。

【図4】第1の実施の形態でのデータの読み出し処理を示すフローチャートである。

【図5】第1の実施の形態でタスクディスパッチしたときの例1の説明図である。

【図6】第1の実施の形態でタスクディスパッチしたときの例2の説明図である。

【図7】第2の実施の形態におけるインターフェース

テーブルを示す図である。

【図8】第2の実施の形態でのデータの書き込み処理を示すフローチャートである。

【図9】第2の実施の形態でのデータの読み出し処理を示すフローチャートである。

【符号の説明】

1, 2 タスク使用中フラグ

71 受信タスク識別用フラグ

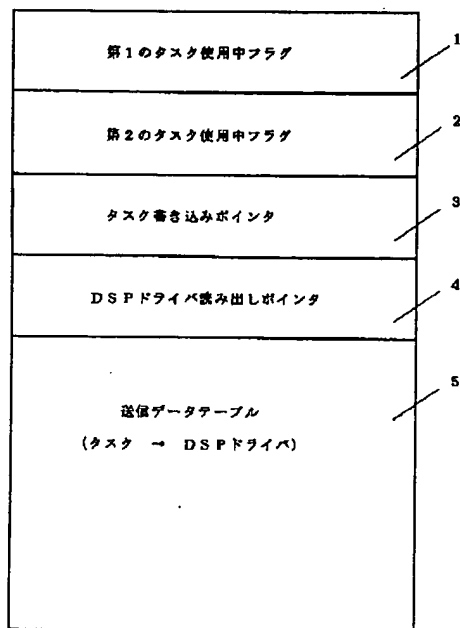
5 送信データテーブル

75 受信データテーブル

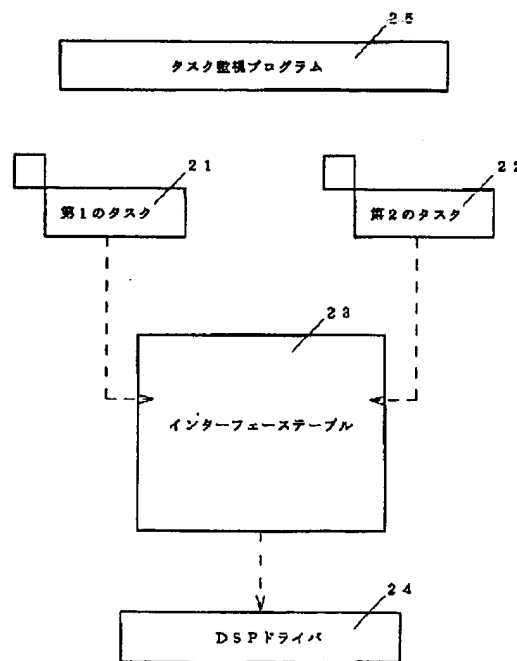
21, 22 タスク

24 DSPドライバ

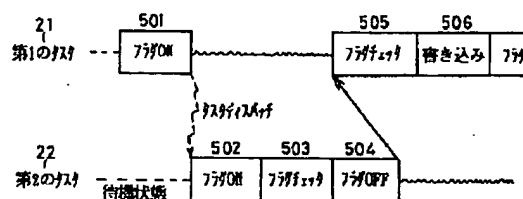
【図1】



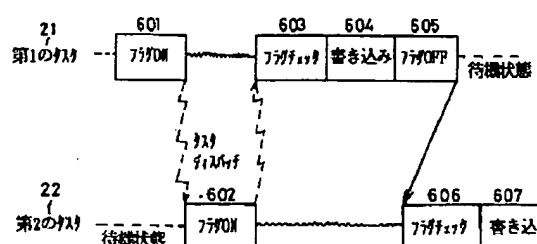
【図2】



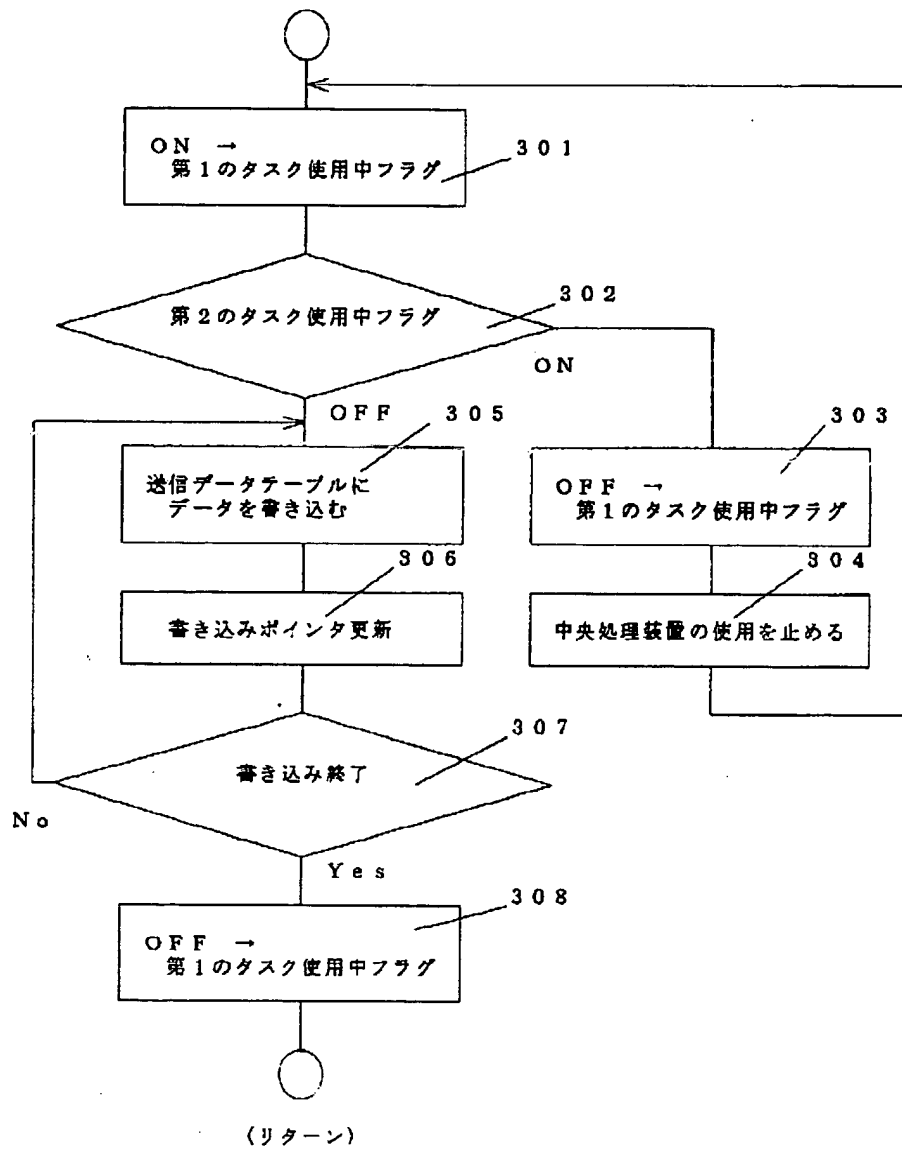
【図5】



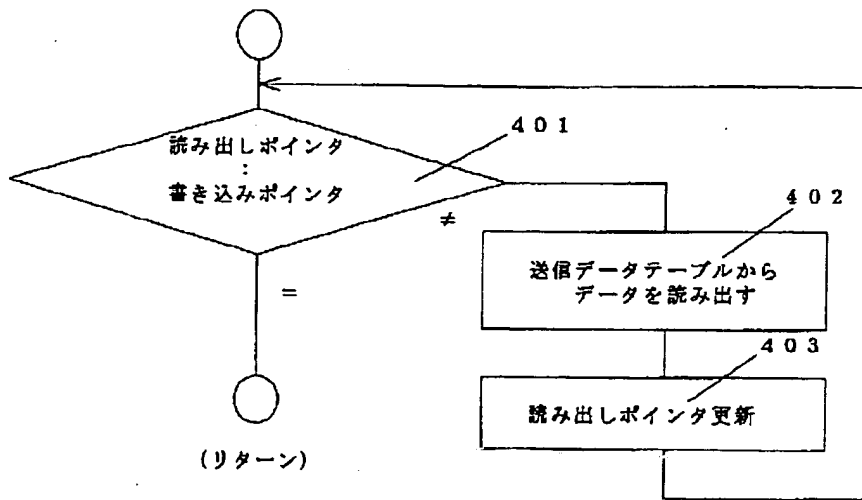
【図6】



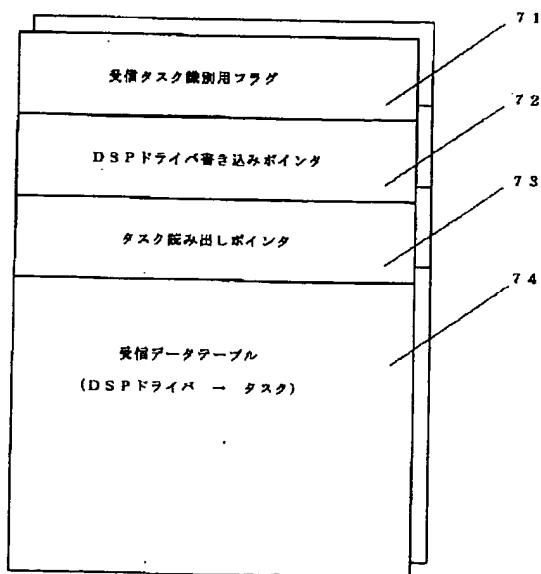
【図3】



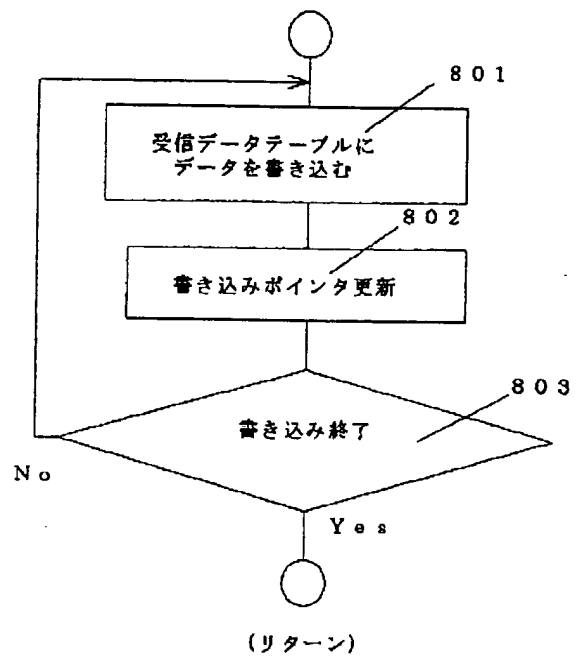
【図4】



【図7】



【図8】





【図9】

